AMENDMENTS TO THE CLAIMS

1. (Currently amended) A computer system with a processing unit and a memory, the processing unit being arranged to fetch memory lines from the memory and execute instructions from the memory lines, each memory line being fetched as a whole and being capable of holding more than one instruction, at least one instruction from the memory lines comprising information, inserted at compile time, that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line, the processing unit being arranged to respond to the information by controlling said part of processing as signaled by the information, said information including at least a fetch bit and a realign bit,

wherein the fetch bit signals explicitly whether or not the subsequent memory line has to be fetched during processing of the instruction, the processing unit being arranged to start fetching of the subsequent memory line in response to the explicit signaling by the fetch bit, and

wherein the ~~information inserted at compile time~~realign bit signals explicitly whether or not an instruction pointer should be updated from a position behind the instruction in the current memory line to a start of the subsequent memory line, so that information following the instruction on the current memory line is skipped over, the processing unit being arranged to update the instruction pointer to the start of the subsequent memory line in response to the ~~information~~explicit signaling by the realign bit.

2. (Canceled).

3. (Canceled).

4. (Canceled).

5. (Previously presented) The computer system according to claim 1, wherein the information signals explicitly whether or not processing of the instruction should be stalled, when the instruction is reached from a branching instruction, processing being stalled to fetch the subsequent memory line that contains a part of the instruction, the processing unit being arranged to stall in response to the information when the instruction is reached from the branching instruction.

6. (Previously presented) The computer system according to claim 1, the processing unit being a VLIW processing unit containing two or more issue slots for issuing operations from the instruction in parallel to different functional units, the instructions being VLIW instructions, capable of containing two or more operations, the instruction comprising a field distinct from the operations to specify said information.

7. (Previously presented) The computer system according to claim 6, the field comprising, in addition to said information, a decompression code that specifies for which issue slots the instruction contains operations.

8. (Currently amended) A method of processing instructions in a computer system with a processing unit and a memory, the processing unit being arranged to fetch memory lines from the memory and execute instructions from the memory lines capable of holding more than one instruction, at least one instruction from the memory lines comprising information, inserted at compile time, that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line, said information including at least a fetch bit and a realign bit,

wherein the fetch bit signals explicitly whether or not the subsequent memory line has to be fetched during processing of the instruction, and wherein the realign bit explicitly signals whether or not an instruction pointer should be updated from a position behind the instruction in the current memory line to a start of the subsequent memory line, the method comprising:

fetching each memory line as a whole, processing an instruction from a current memory line, reading the information from the instruction during processing and controlling said part of

processing as signaled by the information, wherein said controlling comprises starting fetching of the subsequent memory line if fetching is explicitly signaled by the fetch bit, and

~~at least~~ causing a program counter to skip to a start of ~~a~~the subsequent memory line if said skip is explicitly signaled by the realign bit.

Claims 9-15. (Canceled).

16. (Previously presented) The computer system of claim 1 wherein the current memory line and subsequent memory line are positioned adjacent the boundary.

17. (Previously presented) The computer system of claim 1 wherein the fetching of the subsequent memory line consists of fetching a single memory line.

18. (Previously presented) The method of claim 8 wherein the current memory line and subsequent memory line are positioned adjacent the boundary.

19. (Previously presented) The method of claim 8 wherein the fetching of the subsequent memory line consists of fetching a single memory line.

20. (Canceled).